

Definition of AI and a program that satisfies this definition

Dimiter Dobrev
Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
d@dobrev.com

We will consider all possible strategies of the agent and show that one of them is the best. This strategy is not computable, but there are computable strategies close to it. We will define AI as a computable strategy that is close enough to the best. To determine the agent's best strategy, we need a language for description of the world. Through this language we will also make a program satisfying the definition of AI. This program will first understand the world by describing it through the chosen language, then based on this description it will predict the future and choose the best possible action. This program is extremely inefficient and practically unusable, but it can be improved by improving the language for description of the world and by improving the algorithm for predicting the future. In this way, an efficient program satisfying the definition of AI can be obtained.

1. Въведение

Преди време разговарях с колега, който ми каза следното: „Може би, един ден, ние ще създадем AI, но това ще бъде една изключително неефективна програма и за нея ще ни е нужен безкрайно бърз компютър.“ Моят отговор беше: „Ти ми дай тази неефективна програма, която е AI, а аз ще я подобра и ще направя от нея истински AI, който може да работи на реален компютър.“

Днес, в тази статия, аз давам това, което навремето поисках от моя колега. Давам неефективна програма удовлетворяваща дефиницията на AI и дори давам идеи и насоки за това как тази неефективна програма може да се подобри и да стане реална програма работеща в реално време. Надявам се, някой от читателите на тази статия да направи това и да създаде търсения от нас AI.

Доколко неефективна е тук описаната програма? На теория има само два вида програми – завършващи и зацклящи, но на практика някои от завършващите програми са толкова неефективни, че можем да мислим за тях като за зацклящи. Тази програма е такава и поради това е неизползваема (освен ако компютърът е безкрайно бърз или светът е съвсем елементарен).

Каква е дефиницията на AI? Ще дефинираме AI като стратегия. Ако агентът следва тази стратегия, то той ще се справи достатъчно добре. Това важи за произволен свят, в който няма фатални грешки. Ако в света фатална грешка е възможна, тогава агентът може и да не се справи добре в този свят, но средното му представяне върху всички възможни светове пак ще е достатъчно добро.

Кои ще са възможните светове? Стратегиите на света са континуум много. Ако нямаме никакво очакване за това какъв е светът, то няма как да имаме очакване за това какъв е очакваният успех на агента. Ще предположим, че светът може да се опише и че неговото описание е възможно най-просто (това предположение е известно под името „Бръсначът на Окам“). Тоест, избираме един език за описание на светове и се ограничаваме до

световете описани чрез този език. Световете, чието описание е по-просто (по-кратко) ще бъдат предпочитани (ще имат по-голяма тежест).

В тази статия ще разгледаме няколко езика за описание на света. Първият език ще описва детерминирани светове. В този език светът ще се описва чрез изчислима функция, която от състоянието на света и действието на агента изчислява новото състояние на света и следващото наблюдение. В този случай, ако знаем началното състояние на света и действията на агента, получаваме живота на агента в този свят.

Вторият език ще описва недетерминирани светове. Отново имаме изчислима функция. Този път изчислимата функция има още един аргумент и това е случайността. Тук, за да получим живота на агента в този свят ни трябва да знаем още каква е била случайността.

Ще дефинираме AI чрез тези два езика и ще направим предположението, че тези две дефиниции съвпадат. Ще направим дори предположението, че дефиницията на AI не зависи от това кой език за описание на света сме избрали и че всички езици дават една и съща дефиниция на AI.

На базата на тези два езика ще направим две програми, които удовлетворяват дефиницията на AI. Тези две програми изчисляват приблизително една и съща стратегия, но тяхната ефективност е съвсем различна. Тоест, езикът за описание на света не влияе върху дефиницията на AI, но силно влияе върху ефективността на получения чрез този език AI.

Приноси.

Тази статия подобрява дефиницията на AI, която първоначално е създадена от Hernández-Orallo et al. през 1998 [3] и съществено е подобрена от Marcus Hutter през 2000 [4].

Подобренията са две:

1. Дефиниция на AI, която не зависи от сложността на света и дължината на живота.

В [3, 4] е дадена дефиниция на AI, но там се предполага, че сложността на света и дължината на живота са ограничени с някакви константи, които се явяват параметри на дефиницията.

2. Дефиниция на AI, която не зависи от езика за описание на света. В [3, 4] езикът е фиксиран. Тоест, в [3, 4] се предполага, че има само един възможен начин, по който може да се опише света.

2. Постановка на задачата

Нека агентът да има n възможни действия и m възможни наблюдения. Три от наблюденията ще са специални. Това ще са *good*, *bad* и *finish*. Наблюдението *finish* е специално с това, че то е последно (след него не може да има повече действия).

Наблюденията *good* и *bad* са специални с това, че ни дават наградите 1 и -1, наблюдението *finish* също ни дава награда -1, а останалите наблюдения ни дават награда 0.

Забележка: Няма да гледаме на наблюдението *finish* като край на живота, а като скок в неизвестното. Тоест, това ще е моментът, в който моделът се чупи и от този момент нататък не може да предскаже бъдещето. Искаме нашият AI да избягва такива скокове в неизвестното и затова наблюдението *finish* получава награда -1.

Дефиниция 1: Дървото на всички възможности е едно безкрайно дърво. От всеки връх, който е на четна дълбочина и не е листо, излизат n стрелки, на които съответстват n -те възможни действия на агента. От всеки връх, който е на нечетна дълбочина излизат m стрелки, на които съответства m -те възможни наблюдения на агента. Стрелката, която съответства на *finish*, води до листо. Останалите стрелки водят до върхове, които не са листа.

Дефиниция 2: Свят ще наричаме тройката $\langle S, s_0, f \rangle$, където:

1. S – крайно или изброимо множество от вътрешните състояния на света.
2. $s_0 \in S$ – началното състояние на света.
3. $f: S \times \Sigma \rightarrow (\Omega \setminus \{ finish \}) \times S$ – функция, която на състояние и действие връща наблюдение и ново състояние на света.

Функцията f не може да върне наблюдението *finish* (то се наблюдава, само когато f не е дефинирана и когато няма следващо състояние на света). Дали функцията f е изчислима, детерминирана или тотална? Отговорът на всеки един от тези три въпроса може да е Да, а може да е Не.

Дефиниция 3: Стратегия на агента е функция, която на всеки връх (от дървото на всички възможности) ни съпоставя оценка.

Няма да правим разлика между агент и стратегия на агента. За всеки връх от дървото на всички възможности тази стратегия ще ни казва какво действие да изберем. Важни са върховете, които са на нечетна дълбочина (на останалите върховете ще предполагаме, че съпоставяме максимума от оценките на техните преки наследници). Когато стратегията се изпълнява винаги се избира действието, чиято стрелка води към най-голямата оценка. Ако има няколко най-големи оценки, тогава се избира една от тях.

Не искаме стратегията на агента да зависи прекалено силно от оценката, затова ще приемаме две стратегии за еднакви, ако за всеки връх двете стратегии препоръчват едно и също действие (действия).

Дефиниция 4: Живот ще наричаме път в дървото на всички възможности започващ от корена, който или завършва с наблюдение, или е безкраен.

Всеки живот може да се представи чрез редицата от действия и наблюдения:

$$a_1, o_1, \dots, a_b, o_b, \dots$$

Дефиниция 5: Дължината на живота ще бъде t (броят на наблюденията). Тоест дължината на живота ще е дължината на пътя върху две.

Дефиниция 6: Незавършен живот ще е такъв, който може да се продължи. Тоест, това ще е краен живот, който завършва с наблюдение различно от *finish*. Това ще е връх в дървото на всички възможности, който е на четна дълбочина и не е листо.

Когато пушем един агент е в един свят, резултатът ще е живот (завършен). Ако агентът е недетерминиран (няколко най-големи оценки), тогава резултатът не е единствен. Същото, ако светът е недетерминиран.

3. Оценка

Искаме да определим най-добрата стратегия на агента. За целта ни трябва оценка на живота. Тази оценка ще ни даде една линейна наредба, която за всеки два живота ще ни каже кой е по-добрият.

Първо ще кажем какъв е успехът на живота L . Когато животът е краен, ще преброим колко пъти сме наблюдавали *good* – това число ще означим с $L_{good}(L)$. Аналогично за *bad* и *finish*. По този начин успеха за краен живот ще бъде:

$$Success(L) = \frac{L_{good}(L) - L_{bad}(L) - L_{finish}(L)}{|L|}$$

Ще означим с L_i началото на живота L , което е с дължина i . За безкраен живот L ще дефинираме $Success(L)$ като границата на $Success(L_i)$, когато i клони към безкрайност. Ако тази редица не е сходяща, ще вземем средното между точната долна и точната горна граница.

$$Success(L) = \frac{1}{2} \cdot \left(\liminf_{i \rightarrow \infty} (Success(L_i)) + \limsup_{i \rightarrow \infty} (Success(L_i)) \right)$$

По този начин на всеки живот съпоставихме едно число, което е в интервала $[-1, 1]$ и което е успехът на този живот. Дали успехът на живота да не бъде оценката, която търсим? Това не е добра идея, защото, ако в един свят няма фатални грешки, тогава на най-добрата стратегия ще ѝ е все едно какво играе. Ще има един максимален успех и този максимален успех винаги ще може да бъде постигнат, без значение колко грешки са направени в началото. Бихме искали, ако има две възможности, които в безкрайност дават еднакъв успех, най-добрата стратегия да предпочете тази възможност, която по-скоро ще даде положителен резултат. Затова дефинираме оценка на завършен живот по следния начин:

Дефиниция 7: Оценката на безкрайния живот L ще бъде редицата започваща с успеха на живота и продължена с наградите получени на i -тата стъпка:

$$Success(L), reward(o_1), reward(o_2), reward(o_3), \dots$$

Дефиниция 8: Оценката на крайния завършен живот L ще бъде същата редица, но за $i > t$ членовете $reward(o_i)$ ще бъдат заместени с $Success(L)$:

$$Success(L), reward(o_1), \dots, reward(o_t), Success(L), Success(L), \dots$$

(Тоест, наблюденията, които са след края на живота, ще получат очакване за наградата, което е равно на успеха на крайния завършен живот.)

Ще сравняваме две оценки, като вземем първата разлика. Тоест, първата цел на най-добрата стратегия ще бъде успеха на целия живот, но втората цел ще бъде максимално бърз резултат.

4. Най-добрата стратегия

Нека вземем стандартния език за описание на светове. Това е езикът, при който светът се описва с изчислима функция (така е в [3, 4]). За да опишем изчислимата функция f ще използваме Машина на Тюринг. Началното състояние на света ще опишем като крайна дума над азбуката на машината на Тюринг. Получаваме изчислим и детерминиран свят, който в общия случай не е тотален.

Дефиниция 9: Свят със сложност k ще бъде свят, в който:

1. Функцията f се описва с Машина на Тюринг с k състояния.
2. Азбуката на машината съдържа $k+1$ символа $(\lambda_0, \dots, \lambda_k)$.
3. Началното състояние на света е дума с не повече от k букви. Азбуката е $\{\lambda_1, \dots, \lambda_k\}$, т.е. азбуката на машината без празния символ λ_0 .

Тук за три различни неща използваме едно и също k , но няма нужда да използваме различни константи.

Ще кажем, коя е най-добрата стратегия за светове, които са със сложност k (важно е, че тези светове са крайно много). За целта на всеки връх ще съпоставим най-добрата оценка. Трябва да определим най-добрата оценка на върховете, които са на нечетна дълбочина. Нека v_{2t} е връх с четна дълбочина и нека v_{2t+1} е негов пряк наследник. Нека животът, който води от корена до v_{2t} е $a_1, o_1, \dots, a_t, o_t$, а действието, което води от v_{2t} до v_{2t+1} е a_{t+1} . За v_{2t+1} ще видим колко модела със сложност k има той.

Дефиниция 10: Един детерминиран свят е модел на v_{2t+1} , ако в този свят агентът би достигнал до v_{2t+1} , ако извърши съответните действия $(a_1, \dots, a_t, a_{t+1})$.

Дефиниция 11: Най-добрата оценка за v_{2t+1} ще е следното:

1. Когато моделът е един, това ще е оценката на живота, който се получава, ако от този връх нататък се следва най-добрата стратегия.
2. Когато имаме много модели, тогава най-добрата оценка ще е очакването за оценката на живота, който се получава, ако от v_{2t+1} нататък се следва най-добрата стратегия.

Дефиниция 12: Най-добрата стратегия за светове, които са със сложност k , ще бъде следното:

Първи случай. Върхът v_{2t+1} няма модел със сложност k .

Тогава най-добрата оценка на v_{2t+1} ще бъде *undef*. В този случай стратегията няма да знае какво да прави (в цялото поддърво на v_{2t+1}), защото най-добрата оценка на всички наследници ще бъде *undef*.

Ако не искаме да въвеждаме оценка *undef*, можем да използваме най-малката възможна оценка, което е редицата от минус единици. Когато избираме максимума, го избираме между върховете, които не са *undef*. Ако заменим *undef* с възможно най-малката оценка, резултатът ще е същият.

Втори случай. Върхът v_{2t+1} има един модел със сложност k .

Нека D е този модел. В този случай има континуум много пътища минаващи през v_{2t+1} , такива че D е модел на всичките тези пътища. На всеки един от тези пътища (завършени животи) съответства по една детерминирана стратегия на агента. Ще изберем от тези

стратегии множеството на най-добрите. Оценката, която търсим, ще бъде оценката на тези най-добри стратегии.

Ще построим множеството на най-добрите детерминирани стратегии по следния начин: Нека P_0 е множеството на всички стратегии, които водят до v_{2t+1} . Взимаме успеха на всяка една от тях (D е светът). Взимаме подмножеството P_1 от тези стратегии, в които се постига максималният успех. Намаляваме P_1 , като взимаме тези стратегии, при които за $reward(o_{t+2})$ се постига максимум и получаваме P_2 . След това повтаряме процедурата за всяко $i > 2$. По този начин получаваме множеството на най-добрите детерминирани стратегии:

$$P = \bigcap_{i=0}^{\infty} P_i$$

Множество P можем да си го мислим като една недетерминирана стратегия. Нека вземем някое $p \in P$. По този начин ще получим и най-добрата оценка:

$$Success(p), reward(o_{t+1}), reward(o_{p,t+2}), reward(o_{p,t+3}), \dots$$

Тук пропускаме членовете $reward(o_i)$ при $i \leq t$, защото те са определени еднозначно от v_{2t} . Следващият член зависи от v_{2t+1} и D , но не зависи от p . Оставащите членове зависят от p .

Горната формула можем да запишем и по този начин:

$$\max_{p \in P_0} Success(p), reward(o_{t+1}), \max_{p \in P_1} reward(o_{p,t+2}), \max_{p \in P_2} reward(o_{p,t+3}), \dots$$

Трети случай. Върхът v_{2t+1} има краен брой модели със сложност k .

Нека M е множеството на тези модели. Отново имаме континуум много пътища минаващи през v_{2t+1} , такива че за всеки един от тях някой свят от M му е модел. Тези пътища отново образуват дърво, но във втория случай разклоненията бяха само заради различна стратегия на агента, а тук може да има разклонения заради различен модел на света. Отново имаме континуум много детерминирани стратегии, но те вече не съответстват на пътища, а на поддървета, защото може да има разклонение по модела. Отново ще търсим множеството на най-добрите детерминирани стратегии и търсената оценка ще бъде тяхната средна оценка (средната оценка в M).

Отново ще построим множествата от стратегии P_i . Тук P_1 ще бъде множеството от стратегиите, при които се постига максимумът на средния успех. Съответно P_2 ще бъде множеството на тези от тях, при които се постига максимумът на средното на $reward(o_{t+2})$ и т.н. Получената оценка ще изглежда по следния начин:

$$\max_{p \in P_0} \sum_{m \in M} q_m \cdot Success(m, p), \sum_{m \in M} q_m \cdot reward(o_{m,t+1}), \max_{p \in P_1} \sum_{m \in M} q_m \cdot reward(o_{m,p,t+2}), \dots$$

Ако вземем някое $p \in P$, тогава получената оценка ще изглежда така:

$$\sum_{m \in M} q_m \cdot \text{Success}(m, p), \sum_{m \in M} q_m \cdot \text{reward}(o_{m,t+1}), \sum_{m \in M} q_m \cdot \text{reward}(o_{m,p,t+2}), \dots$$

Тук q_i са теглата на световете, които са нормализирани, за да се превърнат във вероятности. В случая предполагаме, че световете имат еднакво тегло. Тоест:

$$q_i = \frac{1}{|M|}$$

■

Това, което описахме, прилича на алгоритъм, но не е алгоритъм, а дефиниция, защото вътре има неизчислими стъпки. Описана стратегия е добре дефинирана, макар и неизчислима. Как чрез най-добрата стратегия за сложност k ще получим най-добрата стратегия за всяка сложност?

Дефиниция 13: Най-добрата стратегия ще бъде границата на най-добрите стратегии за световете, които са със сложност k , когато k клони към безкрайност.

Как ще дефинираме граница на редица от стратегии? За всеки връх оценката ще бъде границата от оценките. Какво ще е граница на редица от оценки? Числото, което е на позиция i ще бъде границата от числата, които са на позиция i . Когато редицата е разходяща, ще вземем средното между точната долна и точната горна граница.

С какво най-добрата стратегия е по-добра от най-добрата стратегия за светове със сложност k ? Първата стратегия знае какво да прави във всеки връх, докато втората стратегия в повечето върхове не знае какво да прави, защото в тях няма нито един модел със сложност k . Първата стратегия може да даде по-добро решение от втората дори и за върховете, за които втората стратегия знае какво да прави, защото първата стратегия разглежда и моделите със сложност по-голяма от k . На пръв поглед не използваме бръснача на Окам, защото всички модели имат еднаква тежест, но всъщност използваме бръснача на Окам, защото по-простите светове се изчисляват от повече машини на Тюринг, тоест те имат по-голяма тежест.

5. Дефиниция на AI

Дефиниция 14: AI ще бъде изчислима стратегия, която е достатъчно близо до най-добрата.

Тук трябва да кажем какво значи една стратегия да е близо до друга стратегия и колко е достатъчно близо.

6. Програма удовлетворяваща дефиницията

Ще опишем алгоритъм, който за всеки незавършен живот ни дава действие, с което този живот да бъде продължен. Алгоритъмът ще се състои от две стъпки:

1. Алгоритъмът ще отговори на въпроса „Какво става?“. Това ще стане като намери първото k , за което незавършеният живот има модел. Също така алгоритъмът ще намери

множеството M (множеството на всички модели на незавършения живот, които са със сложност k). За съжаление, това е неизчислимо. За да стане изчислимо ще търсим ефективни модели със сложност k .

Дефиниция 15: Ефективен модел със сложност k ще бъде свят със сложност k (дефиниция 9), при който машината на Тюринг използва не повече от $1000 \cdot k$ стъпки, за да направи една стъпка от живота (да изчисли следващото наблюдение и следващото вътрешно състояние на света).

Тук числото 1000 е някакъв параметър на алгоритъма, но приемаме, че този параметър не е много съществен. Ако един връх има модел със сложност k , но няма ефективен модел със сложност k , тогава $\exists n (n > k)$ такава, че връхът има ефективен модел със сложност n .

2. Алгоритъмът ще отговори на въпроса „Какво да правя?“. За целта ще проиграем h стъпки в бъдещето по всички модели от M и по всички възможни действия на агента. Тоест, ще обходим едно крайно поддърво и ще изчислим най-добрата частична стратегия.

Дефиниция 16: Частично поддърво на върха v_{2t} по M на дълбочина h ще бъде поддървото на v_{2t} състоящо се от върховете, които са на дълбочина не повече от $2t+2h$ и които имат модел в M .

Дефиниция 17: Частична оценка на живота от v_{2t+2a} до v_{2t+2b} ще бъде редицата:

$$Success(v_{2t+2b}), reward(o_{t+a+1}), \dots, reward(o_{t+b})$$

Забележка: На всеки връх съответства карен живот. Затова, ако v е такъв връх, то $Success(v)$ е успехът на живота съответстващ на v .

Дефиниция 18: Частична стратегия е функция, която на всеки връх от частичното поддърво ни съпоставя частична оценка.

Дефиниция 19: Най-добрата частична стратегия е тази, която на всеки връх ще съпоставя най-добрата частична оценка (от тези, които започват от този връх и завършват с листо).

Дефиниция 20: Най-добрата частична оценка $best$:

1. Ако v_{2t+j} е листо.

1.1. Ако $j=2h$, тогава $best(v_{2t+j}) = (Success(v_{2t+2h}), reward(o_{t+h}))$.

1.2. Ако $o_{t+j/2} = finish$, тогава $best(v_{2t+j})$ е редицата от $Success(v_{2t+j})$ повторено $2+h-j/2$ пъти. (Повторенията са защото искаме на една и съща дълбочина частичните оценки да имат еднаква дължина.)

2. Ако j е четно, тогава:

$$best(v_{2t+j}) = reward(o_{t+j/2}) \text{ insert_at_1_in } \max_{a \in \Sigma} best(v_a)$$

Тоест, взимаме най-добрата частична оценка от преките наследници и я удължаваме с едно, като вмъкваме на първа позиция $reward(o_{t+j/2})$.

Тук с v_a сме отбелязали прекия наследник на v_{2t+j} получен след действието a . Аналогично по-долу за v_o .

3. Ако j е нечетно, тогава:

$$best(v_{2t+j}) = \sum_{o \in \Omega} p_o \cdot best(v_o)$$

Тук с p_o сме означили вероятността следващото наблюдение да е o . Нека с $M(v)$ да означим множеството на моделите на v . Тогава:

$$p_o = \frac{(\sum_{m \in M(v_o)} q_m)}{(\sum_{m \in M(v_{2t+j})} q_m)} = \frac{|M(v_o)|}{|M(v_{2t+j})|}$$

Тук q_m са теглата на моделите. Последното равенство се получава от предположението, че тези тегла са равни. Там където няма модел съответното p_o е нула и няма нужда $best(v_o)$ да се изчислява. ■

Показахме как се изчислява най-добрата частична стратегия. Това ли ще е стратегията на нашия алгоритъм? Отговорът ще е Не, защото искаме да допускаме известен толеранс.

Ако две стратегии се различават незначително по първата координата на оценката си, тогава, ако леко увеличим h , е много вероятно подредбата да се промени. Затова една стратегия ще бъде предпочетена, ако тя е съществено по-добра (тоест, разликата при някоя от координатите е повече от ε).

Ще дефинираме най-добрата частична стратегия с толеранс ε и това ще е стратегията на нашия алгоритъм.

7. Толеранс ε

Ще модифицираме горния алгоритъм като променим функцията $best$. Тази функция връщаше най-добрата оценка, а сега ще връща множеството на най-добрите оценки с толеранс ε .

Ще променим търсенето на максимална оценка в множество от оценки. Преди се гледаше първата координата и се взимаха оценките при които тази координата е най-голяма, после от тях се взимаха тези оценки, при които втората координата е най-голяма и т.н. Накрая се получаваше само една оценка. След модификацията ще вземем тези оценки, при които първата координата е най-голяма и тези които са на разстояние ε от максимума. Нека E_0 е стартовото множество от оценки. Нека с $\varepsilon_0(e)$ сме означили разстоянието до максимума на първата координата $\forall e \in E_0$. Тогава E_1 ще са оценките, за които $\varepsilon_0(e) < \varepsilon$. Нека $\varepsilon_1(e)$ е разстоянието до максимума на втората координата $\forall e \in E_1$.

$$E_1 = \{ e \in E_0 \mid \varepsilon_0(e) < \varepsilon \}$$

$$E_2 = \{ e \in E_1 \mid \varepsilon_0(e) + \varepsilon_1(e) < \varepsilon \}$$

Множеството на най-добрите оценки с толеранс ε ще бъде:

$$E_{h+1} = \{ e \in E_h \mid \varepsilon_0(e) + \dots + \varepsilon_h(e) < \varepsilon \}$$

Трябва да модифицираме и сумата на оценки. Сега оценките са заменени с множества от оценки. Ще направим всички възможни комбинации и за всяка една комбинация ще изчислим сумата. Полученото множество ще бъде множеството от всички суми от всички възможни комбинации.

Остава да изберем следващия ход. Ще вземем множествата от оценки, които ни дава функцията $best$ за преките наследници на v_{2t} . Ще направим обединението на тези множества и ще изчислим множеството на най-добрите оценки с толеранс ε . Ще изберем някое от действията, които ни водят към една от тези най-добри оценки.

8. Дали това е AI?

Дали описаният по-горе алгоритъм удовлетворява нашата дефиниция на AI? Първо ще кажем, че този алгоритъм зависи от параметрите h и ε . За да намалим броя на параметрите ще приемем, че ε е функция на h . Например $\varepsilon = h^{-0.5}$.

Твърдение 1: При достатъчно голямо h описаният алгоритъм е достатъчно близо до най-добрата стратегия.

Не можем да докажем горното твърдение. Дори не сме казали какво значи една стратегия да е „близо“ до друга. Можем да приемем, че когато h клони към безкрайност описаната стратегия клони към най-добрата стратегия за световите, които са със сложност k . Когато t клони към безкрайност k ще достигне сложността на света или ще клони към безкрайност. Тези разсъждения ни карат да си мислим, че горното твърдение е вярно.

9. Свят със случайност

Първият език за описание на светове, който разгледахме, описва детерминирани светове. Ако в света има някаква случайност, тогава описанието на този език ще е много далеч от истината. Ще добавим случайност към езика за описание на светове. По този начин езикът ще стане много по-добър и по-изразителен.

При новия език светът отново ще се описва чрез изчислима функция, но тази функция ще има още един аргумент и това ще е случайността. Случайност ще наричаме резултата от хвърлянето на един зар. Нека сложността на света да е k . Тогава зарът ще има k възможни резултата. Вероятностите да се падне всеки един от тези резултати ще бъдат p_1, \dots, p_k .

Дефиниция 21: Модел на живота до момента t със сложност k ще бъде свят със сложност k и случайност с дължина t . Случайността ще е произволна дума R с дължина t . Буквите на R ще са от азбуката на машината на Тюринг без λ_0 .

Теглото на модела е вероятността да се случи R .

Дефиниция 22: Теглото на модела ще бъде $p_1^{L_{\lambda_1}(R)}, \dots, p_k^{L_{\lambda_k}(R)}$.

Вероятностите p_1, \dots, p_k на модела ще ги определим така, че вероятността да се случи R да е максимална:

$$p_i = \frac{L_{\lambda_i}(R)}{|R|}$$

Тоест, ще имаме модели с малко теглото, при които вероятността да се случи точно този живот е много малка и модели с голямо тегло, при които тази вероятност е по-голяма.

10. Дефиниция със случайност

По същия начин както направихме по-горе ще дефинираме най-добрата стратегия за моделите, които са със сложност k . (Тук е важно, че моделите имат различно тегло.) Ще направим стратегията, която е границата когато k клони към безкрайност и това ще е най-добрата стратегия. Отново дефиницията на AI ще бъде детерминирана стратегия, която е достатъчно близо до най-добрата стратегия.

Твърдение 2: Двете дефиниции на AI съвпадат.

Тоест, най-добрата стратегия за световите без случайност е същата като най-добрата стратегия за световите със случайност. За да докажем горното ни е нужно първо да докажем:

Твърдение 3: Ако имаме произволна дума ω над азбуката $\{0, 1\}$, в която единиците са с вероятност p и ако направим естественото продължение на тази дума, то следващата буква ще е единица с вероятност p .

Какво е естествено продължение? Нека вземем първата (най-простата) машина на Тюринг, която генерира ω . Естественото продължение ще наричаме продължението, което ще се генерира от тази машина на Тюринг.

Не можем да докажем твърдение 3, но ще дадем две идеи за неговото доказателство:

Първата идея е практически експеримент. Ще напишем програма, която намира естественото продължение на редица и ще експериментираме. Ще дадем на програмата различни думи ω , в които 1 е с вероятност p . Ще видим какво е продължението и ще изчислим средната вероятност от тези експерименти. Ако експериментите са много и ако получената средна вероятност е p , можем да предполагаме, че твърдение 3 е вярно.

Втората идея за доказателство е теоретична. Нека имаме една изчислима функция f от \mathbb{N} в \mathbb{N} . Нека сме тръгнали от числото n . Получаваме редицата $f^i(n)$. Ще превърнем тази редица в редицата от нули и единици b_i . Числото b_i ще е нула iff $f^i(n)$ е четно. Нека ω е начало на b_i . Какво е очакването ни за следващия елемент на b_i ?

1 сл. Редицата b_i е циклична и има вида $\omega_1\omega_2^*$. Нека ω е по дълга от ω_1 . Тогава има начало на ω_2 което е част от ω и за него единиците са с вероятност p .

2 сл. Редицата $f^i(n)$ има дълго начало, в което нечетните числа са с вероятност p . Няма причини да очакваме тази вероятност да се промени.

11. Програма със случайност

Ще направим програма, която удовлетворява дефиниция на AI, която е направена на базата на модели със случайност. Ще направим тази програма по същия начин, както я направихме по-нагоре, но ще има някои разлики.

Няма да търсим първото k , за което има модел до момента t със сложност k , защото такъв модел ще има дори и при съвсем малко k . Вместо това, ще предполагаме, че k е фиксирано и че k е параметър на алгоритъма.

Ще обикаляме на дълбочина h частично поддърво на върха v_{2^t} по всички модели със сложност k , по всички възможни действия на агента и по всички случайности $R_1 R_2$, където R_1 е случайността до t , а R_2 е случайността след t . Тук R_1 е фиксирана (определя се от модела), а R_2 пробягва всички възможности.

Аналогично на твърдение 1 ще направим:

Твърдение 4: При достатъчно големи k и h описаният алгоритъм е достатъчно близо до най-добрата стратегия.

Твърдим, че при достатъчно големи параметри двата алгоритъма изчисляват приблизително една и съща стратегия, но дали тези два алгоритъма са еднакво ефективни?

На практика и двата алгоритъма са безкрайно неефективни, но, все пак, вторият алгоритъм е много по-ефективен от първия. Ще разгледаме три случая:

1. Нека имаме прост детерминиран свят. Прост в смисъл, че е със сложност k , където k е малко. В този случай първият алгоритъм ще е малко по-ефективен, защото бързо ще намери модела. Вторият алгоритъм ще намери същия модел, защото детерминираният модел са подмножество на недетерминираният.

2. Нека имаме детерминиран свят, който не е прост, т.е. той е с голямо k . В този случай на първия алгоритъм ще му е нужно ужасно много време за да намери модел на света. Вероятно, той няма да намери истинския модел на света, а едно по-просто обяснение. Това по-просто обяснение ще е модел на живота до t , но след още няколко стъпки модела ще сгреша. Вторият алгоритъм също ще намери по-просто обяснение на света, но това по-просто обяснение ще е недетерминирано. И двата алгоритъма ще предсказват бъдещето с някаква грешка, но описанието, което включва случайност ще е по-добро и по-точно. Освен това, сложността на описанието със случайност ще е много по-малка.

3. Нека имаме свят със случайност. В този случай вторият алгоритъм има съществено преимущество. Той ще намери недетерминирания модел на света и ще започне да предсказва бъдещето по най-добрия възможен начин. На пръв поглед изглежда сякаш първият алгоритъм въобще няма да се справи, но това не е така. И той ще се справи, макар и много по-бавно. Недетерминирания модел се състои от изчислима функция f и случайност R . Съществува изчислима функция g , която генерира R . Композицията на f и g ще бъде детерминиран модел на света в момента t . Разбира се, след още няколко стъпки g ще се размине от действителната случайност и $f \circ g$ вече няма да е модел на света. Тогава ще трябва да търсим нова функция g . Тоест, възможно е чрез детерминистична функция да

опишем свят със случайност, но това описание е много измъчено и работи само до някой момент t . Недетерминистичният модел ни дава описание, което работи за всяко t .

Извода е, че изборът на език за описание на света е много важен. Може би тези два езика дават една и съща дефиниция на AI, но програмите получени на базата на двата езика имат съществено различна ефективност.

12. Свят с много агенти

Светът със случайност може да се разглежда като свят с един агент, който играе случайно. Нека допуснем, че светът има много агенти и те са разделени на три вида:

1. Приятели, които ни помагат.
2. Врагове, които се стремят да ни попречат.
3. Такива, които играят случайно.

Нека броят на агентите е g (без да броим главния герой). Ще допуснем, че всеки агент има k възможни хода (k е сложността на света). Ще предполагаме, че първи играе главния герой (това сме ние), а след него играят останалите агенти, като реда по който играят е фиксиран. Предполагаме, че всеки агент вижда всичко (вижда вътрешното състояние на света, вижда модела, включително броя на агентите и това кой е приятел и кой е враг и вижда как вече са играли тези агенти, които играят преди него). Ще предполагаме още, че агентите са много умни и могат да изчислят кой е най-добрият и кой е най-лошият ход.

Моделът на света отново ще бъде машина на Тюринг, но с повече аргументи (вътрешното състояние на света и хода на главния герой плюс ходовете на всички останали агенти). Моделът ще включва още и това кой ни е приятел и кой ни е враг. Моделът на живота до момента t ще включва още и ходовете на всичките g агента за всичките t стъпки.

Отново ще направим дефиниция на AI на базата на този нов, по-сложен език. Отново ще направим предположението, че това е същата дефиниция. Ще направим и програма, която търси модел на света в множеството на световите с много агенти. Ще видим, че сега езика е много по-изразителен и че ако в света имаме поне един опонент, то този начин за описание на света е много по-адекватен и съответно програмата, която сега сме направили е много по-ефективна.

13. Заключение

Разгледахме три езика за описание на света. На базата на всеки един от тях направихме дефиниция на AI и предположихме, че в трите случая дефиницията е една и съща. Ще направим едно още по-силно предположение:

Твърдение 5: Дефиницията на AI не зависи от езика за описание на света, на базата на който сме направили тази дефиниция.

Това твърдение не можем да го докажем, макар че предполагаме, че то е вярно. Също така предполагаме, че то не може да бъде доказано (подобно на тезиса на Чърч).

Въпреки, че предположихме, че дефиницията на AI не зависи от езика за описание на света, предполагаме, че от този език силно зависи програмата, която удовлетворява тази дефиниция. Видяхме, че при сравнението между първите два езика се видя, че втория е много по-експресивен и получения чрез него AI е много по-ефективен.

Ще разгледаме още един език за описание на светове. Това е езикът описан в [2]. В този език се дефинира понятието „алгоритъм“ и чрез това понятие много по-ефективно се описва света. С понятието „алгоритъм“ ние можем да планираме бъдещето. Нека вземе като пример следното: „Ще чакам автобуса, докато дойде. После ще отида на работа и ще седя там до края на работното време.“ Тези две изречения описват бъдещето чрез изпълнение на алгоритми. Ако ще предвиждаме бъдещето само като обхождаме h възможни стъпки, то това h ще трябва да стане неприемливо голямо.

Езикът описан в [2] е много по-експресивен и той дава надежда, че чрез него ще може да се направи програма, която да е AI и която да е достатъчно ефективна, за да може да работи в реално време.

14. Благодарности

Искам да благодаря на моя колега Иван Сосков. Той е човекът, с когото преди години обсъждах ефективността на AI. Искам да благодаря и на моите колеги Димитър Димитров и Йоан Карадимов. Идеята за тази статия възникна по време на един разговор с тях. Искам да се извиня на Marcus Hutter и Shane Legg, защото в статията си [1] несправедливо ги обвиних, че са използвали моя статия без да я цитират. По-късно, когато внимателно прочетох статията на Marcus Hutter [4], разбрах че там той е публикувал основните идеи много преди мене.

References

[1] Dobrev D. (2013) Comparison between the two definitions of AI. *arXiv:1302.0216 [cs.AI]*

[2] Dobrev D. (2020) Language for Description of Worlds. *arXiv:2010.16243 [cs.AI]*

[3] Hernández-Orallo, J., & Minaya-Collado, N. (1998). A formal definition of intelligence based on an intensional variant of Kolmogorov complexity. *Proc. intl symposium of engineering of intelligent systems (EIS'98), February 1998, La Laguna, Spain (pp. 146–163). : ICSC Press.*

[4] Hutter, M. (2000). A Theory of Universal Artificial Intelligence based on Algorithmic Complexity. *arXiv:cs.AI/0004001 [cs.AI]*